

Amendments to the Claims:

The following listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for performing morphology analysis of a natural language document, comprising the steps of:
 - inputting the natural language document as an input text,
 - tokenize ~~said-the~~ input text, thereby producing a token stream of a plurality of tokens,
 - ~~checking for comparing~~ each token of ~~said-the~~ token stream to a list of previously encountered tokens in the token stream,
 - determining whether it-the token either corresponds to at least one previously encountered token in the list as a recurring token in the token stream, or else is a unique token which is occurring for the first time in the token stream or a recurring without corresponding to the at least one previously encountered token which already occurred earlier in the token stream list,
 - adding the unique token to the list of previously encountered tokens,
 - marking the unique tokens-token with an identification-~~ID~~ and,
 - adding to recurring tokens a pointer directed towards the ID which was defined identification for the ~~respective recurring~~ token ~~when occurring for the first time,~~
 - performing a morphological look-up only on the unique ~~tokens-token~~, thereby producing ~~results of a~~ morphological look-up result,
 - storing said ~~results of~~ morphological look-up result for the unique ~~tokens~~ token together with the ~~ID~~ identification,
 - reading the ~~results of~~ morphological look-up result for the recurring ~~tokens~~ token,
 - joining all results of the morphological look-up results for the unique and recurring tokens, thereby producing a stream of morphological analyses, and
 - outputting ~~said-the~~ stream of morphological analyses.
2. (Currently Amended) The method according to claim 1, wherein the step of storing ~~said results of the~~ morphological look-up result for the unique ~~tokens-token~~ together with the ~~ID~~ identification comprises the step of creating a dynamically extending database.

3. (Original) The method according to claim 2, wherein said dynamically extending database is a self-extending hash table.
4. (Original) The method according to claim 1, wherein the step of tokenizing is performed by a first finite state transducer.
5. (Original) The method according to claim 4, wherein the first finite state transducer includes punctuation conventions and higher level lexical information.
6. (Original) The method according to claim 4, wherein the step of morphological look-up is performed by a second finite state transducer.
7. (Currently Amended) A system for performing morphology analysis of a natural language document, comprising:
 - a tokenizer for tokenizing an input document, thereby producing a token stream of a plurality of tokens,
 - a pre-processor that ~~checks for~~ compares each token of ~~said the~~ token stream to a list of previously encountered tokens in the token stream, determines whether it the token either corresponds to at least one previously encountered token in the list as a recurring token in the token stream, or else is a unique token which is occurring for the first time in the token stream or a recurring without corresponding to the at least one previously encountered token which already occurred earlier in the token stream list, adds the unique token to the list of previously encountered tokens, marks the unique tokens token with an identification (ID) and adds ~~to recurring tokens~~ a pointer directed towards the ~~ID which was defined~~ identification for the ~~respective recurring token when occurring for the first time~~,
 - a morphological look-up module for performing a morphological look-up only on the ~~unique tokens token~~ token, thereby producing ~~results of a~~ morphological look-up result,
 - memory for storing said ~~results of morphological look-up result~~ result for the ~~unique tokens token~~ token together with the ~~ID~~ identification,
 - a post-processor that ~~detects tokens carrying said~~ replaces a token that carries the pointer and replaces them by said results of with the morphological look-up result stored in the memory under the respective ~~ID~~ identification.
8. (Original) The system according to claim 7, wherein the tokenizer is a first finite state transducer.
9. (Original) The system according to claim 8, wherein the tokenizer includes punctuation conventions and higher level lexical information.

10. (Original) The system according to claim 8, wherein the morphological look-up module is a second finite state transducer.

11. (Original) The system according to claim 7, wherein the memory is a dynamically extending database.

12. (Original) The system according to claim 11, wherein said dynamically extending database is a self-extending hash table.

13. (Currently Amended) A system for performing morphology analysis of a natural language document, comprising:

a tokenizer for tokenizing an input document, thereby producing a token stream of a plurality of tokens,

a morphological look-up module for ~~performing the morphological look-up, thereby~~ producing results of a morphological look-up,

memory for storing ~~said the~~ results of the morphological look-up for ~~the a~~ unique ~~tokens-token~~ token of the plurality of tokens together with ~~the ID~~ an identification,

a control unit for controlling ~~said the~~ morphological look-up module and ~~said the~~ memory; wherein ~~said the~~ control unit:

~~checks for compares~~ each token of ~~said the~~ token stream to a list of previously encountered tokens in the token stream, determines whether it the token either corresponds to at least one previously encountered token in the list as a recurring token in the token stream, or else is a the unique token which is occurring for the first time in the token stream or a recurring corresponding to the at least one previously encountered token which already occurred earlier in the token stream list,

adds the unique token to the list of previously encountered tokens, and

marks the unique tokens-token with an the identification-(ID) and,

adds to recurring tokens a pointer directed towards the ID which was defined identification for the respective recurring token when occurring for the first time,

initializes a the morphological look-up for the unique tokens-token, and

replaces the recurring tokens-token marked with said the pointer by the results of the morphological look-up stored in the memory under the respective ID identification.

14. (Original) The system according to claim 13, wherein the tokenizer is a first finite state transducer.

15. (Original) The system according to claim 14, wherein the tokenizer includes punctuation conventions and higher level lexical information.

16. (Original) The system according to claim 14, wherein the morphological look-up module is a second finite state transducer.

17. (Original) The system according to claim 13, wherein the memory is a dynamically extending database.

18. (Original) The system according to claim 17, wherein said dynamically extending database is a self-extending hash table.

19 (New) The method according to claim 1, wherein the step of joining the morphological look-up results further comprises replacing the each token and the recurring token with the pointer from the morphological look-up unless the each token is marked with the pointer.

20 (New) The system according to claim 7, wherein memory stores mapping between the unique token and the identification.